

Complexity and Safety (FAA)

SEI Board of Visitors

October 27, 2016

Sarah Sheard, Team Lead

Team: Mike Konrad, Chuck Weinstock,
Bill Nichols, Greg Such

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213



Copyright 2016 Carnegie Mellon University

This material is based upon work funded and supported by Federal Aviation Administration under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center sponsored by the United States Department of Defense.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of Federal Aviation Administration or the United States Department of Defense.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

DM-0004166

Objectives

Introduce the FAA research task

Show how we tied complexity to safety

Emphasize potential uses and new challenges



Key Topics

Introduction: why we did this

Relationship between complexity and safety
algorithm and example referenced

Applications and new challenges

Introduction



2014: FAA requested research on definition and measurement of complexity for the context of safety assurance

Funded SEI to do a two-year research project

This is the outcome.



Complexity Is Complex

What is “complex”

Size (number)

Diversity

Rips / j

Diversity
interconn

How complex is it?

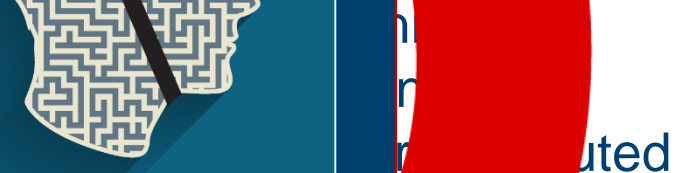
Thematic complexity

an-in
burn

Safety?



Program
(ut)?

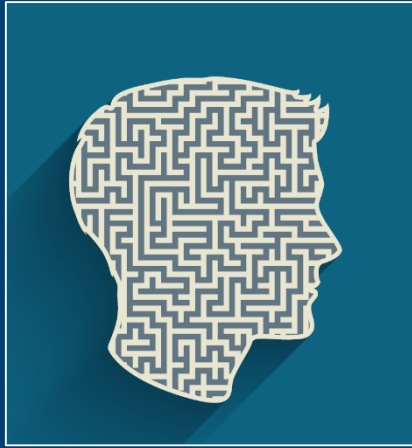


Test

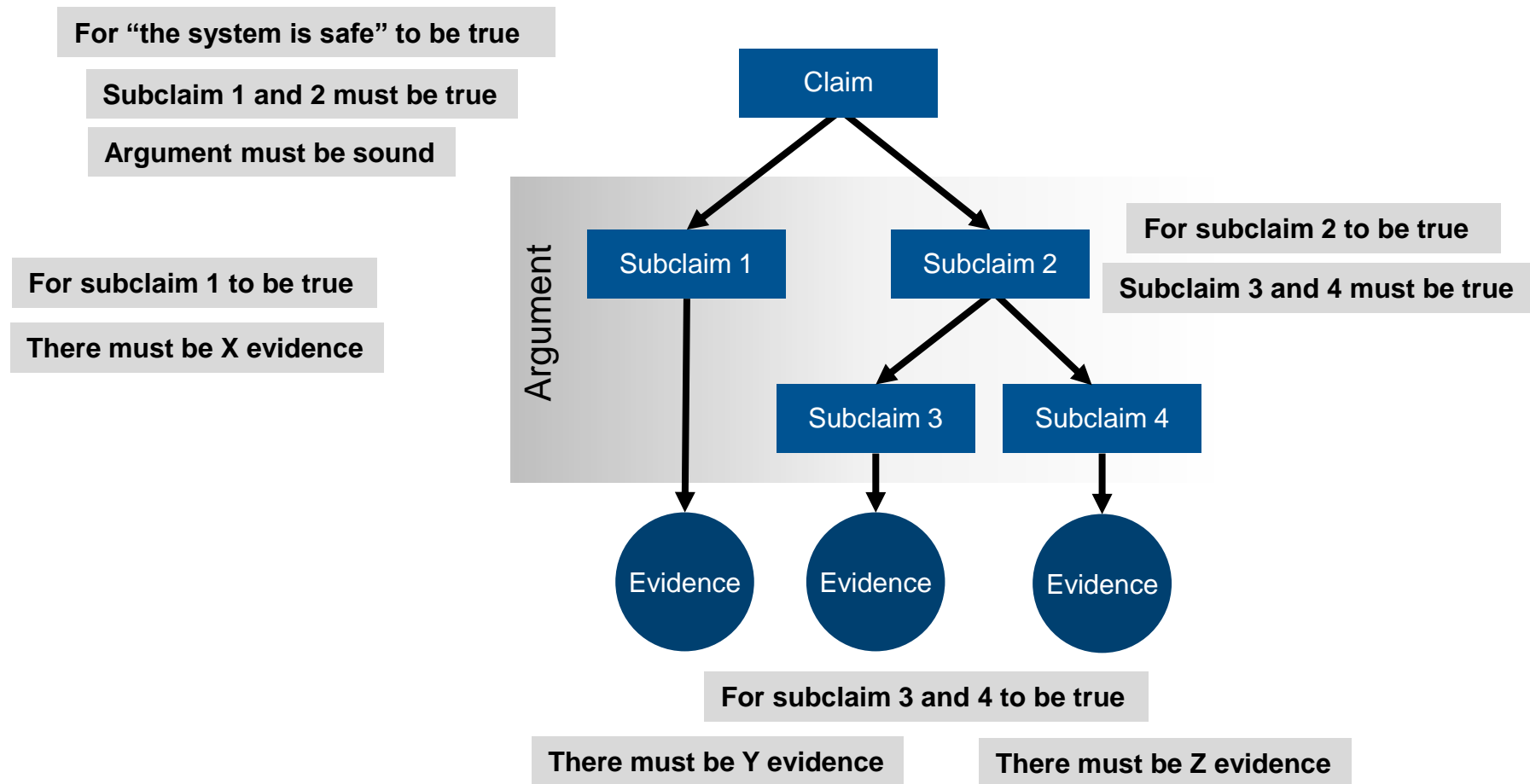
system/
test level?

Complexity

What about matters to



Safety Case (Type of Assurance Case)



Two Breakthroughs

1. Evaluate the **complexity *of the safety case***

But the safety case isn't "complete" until the aircraft is designed, built, and tested, with all software on board...

2. **Estimate** the size of the safety case **early**

How much work (analysis, documentation, meetings etc.)
will it take to prove the system is safe?

(How many potentially-cascading error conditions are there?)

- ☞ Order of magnitude probably OK
- ☞ Assume component assurance process will remain as-is
- ☞ Big open question is errors cascading from one component to another

Used # of safety case arguments as a proxy for complexity



Our Method

Assume the early design work on the new system has resulted in a model of the system architecture at a high level including

- system modes
- active components and their interconnections in each mode
- possible failure conditions that could propagate outward

Use this design to count the number of ways an error could propagate out from the originating component to another component. Determine **error propagation complexity**.

Estimate time and cost for demonstrating safety: multiply *error propagation complexity* by the typical amount of time it takes to understand that one error propagating from one component to another does not cause safety related hazards.

Potential Applications of This Research

FAA uses the research as evidence that they need to ask manufacturers to provide documented safety cases rather than just standards compliance

Manufacturers (first and lower tiers) use estimate of design complexity to estimate their own quality assurance efforts

Enables comparison of designs by how complex their error propagation potentials are

Enable use of complexity as an indicator of risk, to be tracked using standard techniques

Future research into “How much can we discount the complexity of a system given that X% has been used before?” can be framed as “credit for precedence” and ties to “recertification” questions. Much interest across SEI and at CMU for this topic.

#1 Recommended Future Research: Precedence

Study complexity “discounts” that we should give to known or preceded system components because they are familiar

- How many error propagations (from model) have already been proven not to be unsafe and thus need less review?
- How can this be applied to, say, *slightly* different configurations? How do you measure “slightly”?
- How can this be applied to slightly different hazards?
- What is the safety effect of a higher-capability component compared to existing?

Other areas can contribute:

- How organizations today currently allow credit for testing already done
 - FAA and aircraft re-certification (e.g., longer fuselage)
 - FDA and medical devices
 - Regression testing
- Estimate of the amount of impact caused by a change (hardware, then software)
- Understanding how much of the problem could be solved by nearly-independent, modularized, proven-correct components

Other Recommended Future Research

- 1) Apply and validate to larger system at real-life scale.
- 2) Study special cases, assumptions, and limitations more specifically.
 - a) Including tweak numbers for whether the applicant has provided an organized assurance case or not. How does this affect FAA effort?
 - b) Determine effect of having models to different levels of detail. Is there a notional “complexity reduction” curve?
- 3) Expand fault model to include more than error propagation: emergent behavior, concurrency, and cybersecurity.
- 4) Develop guidelines for safe assurance practices and design guidelines to reduce software complexity.



Contact Information

Sarah Sheard

Senior Engineer

Telephone: +1 412.268.7612

Email: sheard@sei.cmu.edu



Backup Slides



Our Method

Primary assumption:

Early design work on a new system* has resulted in a model of the system architecture at a high level, including

- system modes
- active components and their interconnections in each mode
- possible failure conditions that could propagate outward

Many additional assumptions made to arrive at notional thresholds for OK systems or systems too complex to assure as safe

*Future research can address system upgrades and not just new systems.

Our Method, Continued

Estimate size of safety case.

Assume: Size of safety case for ultra-complex systems will be dominated by tracking down potential consequences of each error that could propagate from one component to another.

Assume: Applicant has done FHA,* identified failure conditions that can arise in each component, and how the effects might propagate.

Question: How many opportunities does our system have for that to happen? Becomes “error propagation complexity” = EPC

*FHA = functional hazard analysis

Algorithm

Sum over all system modes:

Sum over all components active in a given mode:

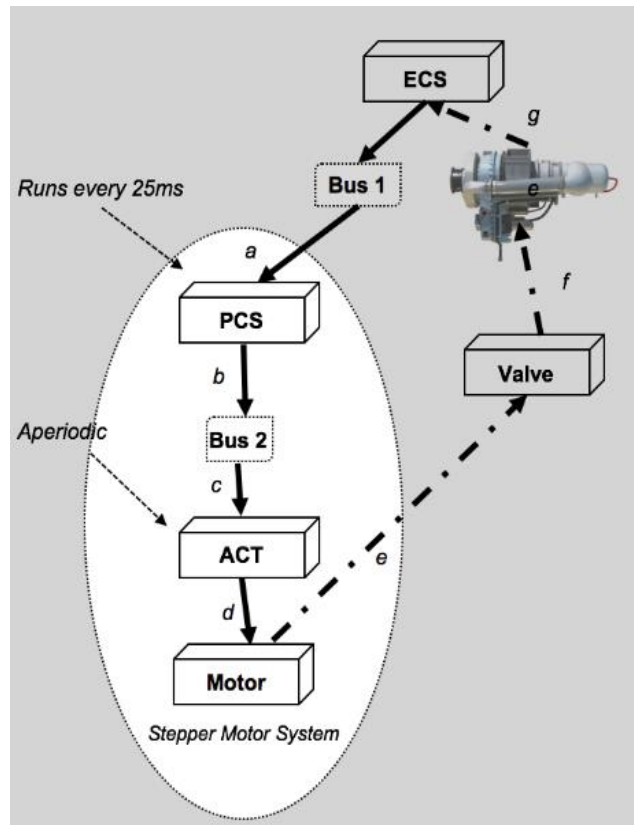
Sum over all propagation points (p-points) for this component:

of:

$\left\{ \begin{array}{l} \text{Number of failures} \\ \text{that could propagate} \\ \text{out from this p-point} \end{array} \right\}$ times $\left\{ \begin{array}{l} \text{Fan-out from} \\ \text{this p-point} \end{array} \right\}$



Example 1: Stepper Motor System



1. From high level design:

- 1 mode
- Interfaces shown
- Treat bus 2 as a component*
- 4 components plus environment
- #P-points = 1 for all components
- Fan-out always = 1

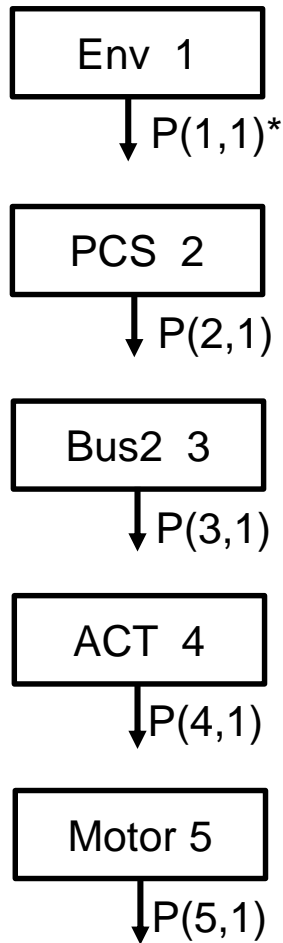
2. From error model:

- Errors from environment to SMS: 3
- Errors from PCS to bus 2: 4
- Errors from bus 2 to ACT: 3
- Errors from ACT to motor: 3
- Errors from motor to envt.:3

Ref: Konrad 2015b of Final Report
*Since it can be a source of a failure condition

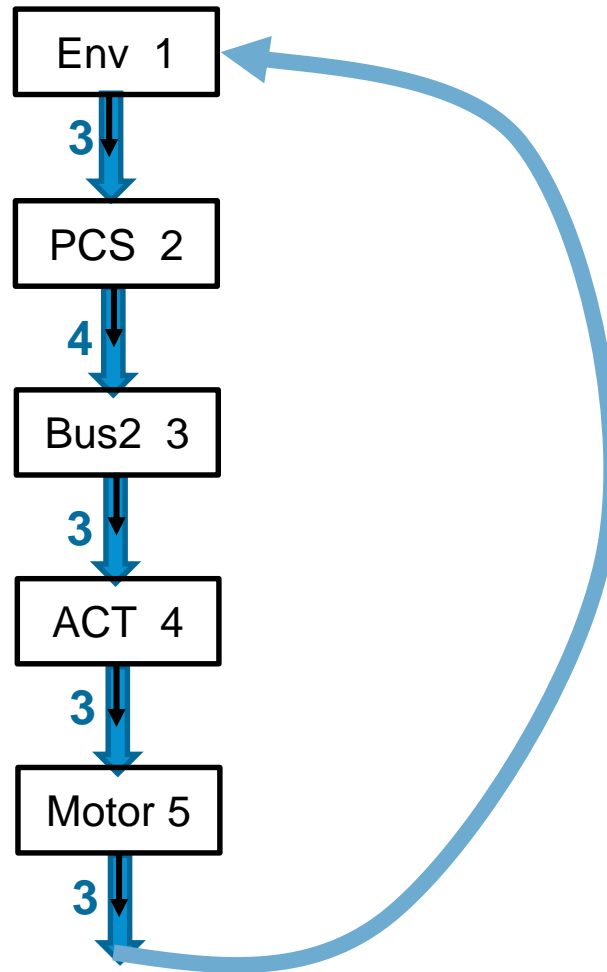
Calculating EPC (for one mode)

First step



*Notation P(component#, p-point#)

Second step



Third step

Sum of (**#failures***fan-out for all P-points of component x)

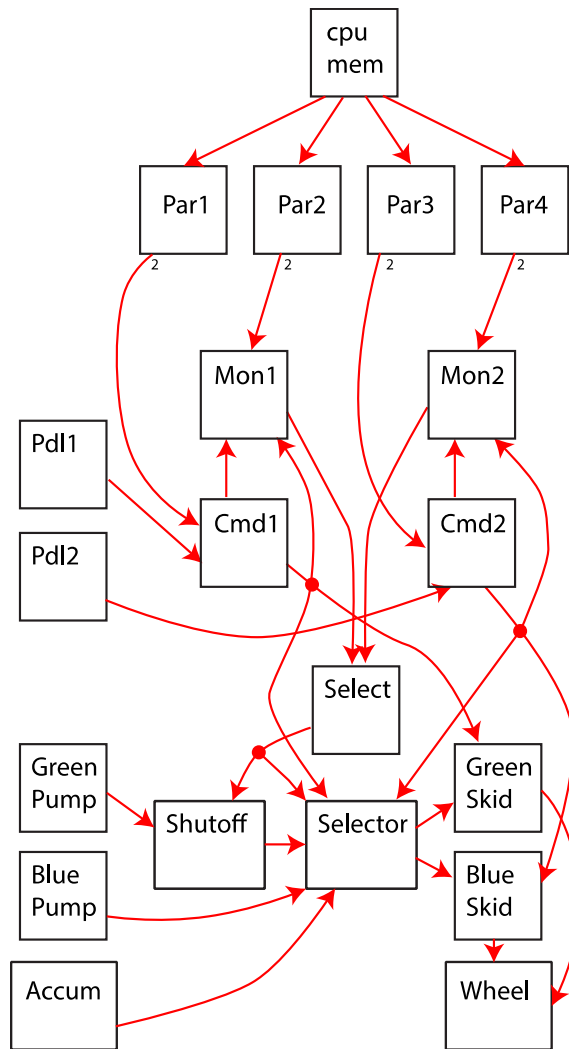
x	Sum
1	$3*1 = 3$
2	$4*1 = 4$
3	$3*1 = 3$
4	$3*1 = 3$
5	$3*1 = 3$

Total all components

Error propagation complexity =

16

Example 2: Wheel Brake System



1. Obtain architecture & fault models for the system
2. Simplify interconnections. Identify # components, # p-points, and fan-out from each of them
3. Identify # of potential errors associated with each propagation point

Ref: Konrad 2016

Wheel Brake System Complexity: Solution

Component Type	Number of Components	Number of P Points per Component	#FC	Fan-Out	#Components * #P points * #FC * FO
cpu mem	1	4	1	1	4
Par1..Par4	4	1	2	1	8
Mon1..Mon2	2	1	1	1	2
Cmd1..Cmd2	2	1	1	1	2
Pdl1..Pdl2	2	1	1	3	6
Select	1	1	1	2	2
Green pump, Blue pump, Accum	3	1	1	1	3
Shutoff	1	1	1	1	1
Selector	1	2	1	1	2
Green skid, Blue skid	2	1	1	1	2
Wheel	1	0	n/a	n/a	0
Error Prop. Complexity:					34

FC = Failure conditions = errors

Maximum Allowable Complexity

What EPC would a system have if the effort it will take to assure that the system is safe would exactly equal all the resources that the FAA has to do so? (using 787 as an example)

$$\text{Effort to assure the system} = \text{Effort to resolve a typical error propagation event} * \text{\# EP events (= EPC)}$$

Or: $\text{EPC} = \text{Effort available} / \text{effort for typical event}$

For 787, FAA-related assurance effort available ~ 100 SY = 12.5 full time staff * 8 yrs

What is effort to resolve a typical error propagation event?

Effort to Resolve Typical Error Prop. Event

Assumption: effort (per safety case statement) ~ code review effort (per statement) (from SEI TSP experience)

=> Worst case 1.31 minutes/error propagation

=> Best case 0.51 minutes/error propagation

Defined hypothetical avionics system (combination of small, medium, and large systems, in hierarchy)

Total review time = 12.2 staff years (best case)
to 39.9 staff years (worst case)

Total complexity for this system = 5110

Ratio: best case:	418.9 error propagation events / SY
worst case:	128.1 error propagation events / SY

*Even though FAA does not do code reviews, these are a better guess at effort than nothing.

Compare to: All The Resources FAA Has

787: certification effort = 100 staff years over 8 years

=> Assume half* (50 SY) related to avionics

Best case complexity = 50 SY * 418.9 EP's/SY = 21,000 EPC

Worst case complexity = 50 SY * 128.1 EP's/SY = 6400 EPC

**(Best case: a more complex system
can be reviewed in the same time)**

Conclusion:

**A system that exactly consumes all review
resources of the FAA would have a complexity
between 21000 (best case) and 6400 (worst case)**



Effort to Resolve a Typical Error Prop. Event

Assumption: resolve EP events in context of reviewing safety (assurance) case provided by applicant

Looked at two ways of estimating

- 1) Time to inspect code, per page (lots of data, not so relevant)
(from SEI Team Software Process experience) = “SCI” rate
- 2) Time to review safety case, per “node” (relevant, less data)
(researcher tested each design twice) = “SCR” rate

1) SCI: 0.94 minutes/event to 1.68 minutes/event, mean 1.31

=> Worst Case 1.31 minutes/event

2) SCR: 0.36 minutes/event to 0.65 minutes/event, mean 0.51

=> Best Case 0.51 minutes/event

*Even though FAA does not do code reviews, these are a better guess at effort than nothing.

Define Hypothetical Avionics System

Suppose the system contains

- 100 small-sized subsystems (~stepper motor)
- 30 medium-sized subsystems (~wheel brake)
- 10 large-sized subsystems (~ hypothetical SAAB-EII 100)

Estimate EPC for each kind of subsystem as follows

- For small- and medium-sized subsystems, we have examples, each of which has two distinct designs => mean EPC:
EPC (small) = 16.5
EPC (medium) = 32
- For the large system, used the example in [Peterson 2015]: hypothetical SAAB-EII 100 aircraft; estimated the EPC based on the top-level system design diagram (min 200, max 300, mean 250)



Assumptions About Modes and Hazards (Claims That Will Need to be Argued)

Based on

Their FHA identifies ~60 system-level hazards (not all of them active in all system modes), we estimate ~30 are relevant to each subsystem in any given mode.

Also assume the level of effort required to consider additional modes is ~twice the effort required to review a subsystem design for a single mode.

*There are both primary system modes (e.g., take-off and climb; in-flight; approach and land) and system sub-modes that need to be considered.

Review Time for Hypothetical Avionics System

Total review time = times(per hazard) * hazards * # systems

Assume 30 relevant hazards times 2 modes = 60

Size	Best case min	Worst case min	Hazards	#	Total Mins
Small	8.1	22.2	60	100	48,600-133,300
Med	15.2	44.4	60	30	27,300-80,000
Large	101.3	392.1	60	10	60,700-235,300

Sum: 137,000-449,000 minutes or 3.0-10.0 years

For team of 4: 12.2 – 39.9 staff years*

*Dedicated to conducting rigorous 4-person reviews 3 hours/day

Complexity for This System

So what system complexity resulted in this amount of resources?
Add up complexity of all systems:

Size	Complexity each	#	Total
Small	16.5	100	1650
Med	32	30	960
Large	250	10	2500
Total			5110

From previous chart, this hypothetical system with complexity **5110** needs **12.2** (best case) to **39.9** (worst case) staff years

=> What complexity of system would use *all the review resources* of the 787 program?



Complexity of “Borderline” System: One That Can Be Reviewed in 787-amount of Resources

Hypothetical system of complexity **5110** needs **12.2** (best case) to **39.9** (worst case) staff years

787 had 100 staff years over 8 years; assume **half** was on avionics (**50** staff years)

What complexity of system would require exactly as many resources to assure as the 787 used?

Best case complexity = $50 * 5110 / 12.2 = \sim 21,000$

Worst case complexity = $50 * 5110 / 39.9 = \sim 6400$

A system that exactly consumes all review resources of the FAA would have a complexity between 21000 (best case) and 6400 (worst case)

